


COMPUTATIONALLY EFFICIENT NOISE REDUCTION FILTER

by
Gopal B. Avinash
New Berlin, Wisconsin

EXPRESS MAIL MAILING LABEL	
NUMBER:	EK 229 729 604 US
DATE OF DEPOSIT:	November 21, 2001
<i>Pursuant to 37 C.F.R. § 1.10, I hereby certify that I am personally depositing this paper or fee with the U.S. Postal Service, "Express Mail Post Office to Addressee" service on the date indicated above in a sealed envelope (a) having the above-numbered Express Mail label and sufficient postage affixed, and (b) addressed to the Assistant Commissioner for Patents, Washington, D.C. 20231</i>	
<i>11/21/01</i>	
Date	Signature

COMPUTATIONALLY EFFICIENT NOISE REDUCTION FILTER

5

FIELD OF THE INVENTION

This invention relates to discrete picture element or pixel imaging techniques, and, more particularly, to an improved technique for analyzing and modifying values representative of pixels in such images which significantly increases computational efficiency while maintaining overall image quality.

10

BACKGROUND OF THE INVENTION

15

A variety of discrete pixel imaging techniques are known and are presently in use. In general, such techniques rely on the collection or acquisition of data representative of each discrete pixel composing an image matrix. Examples of such imaging techniques exist, including optical character recognition, facial feature recognition, corneal scanning, fingerprint recognition, and virtually any other form of digital imaging which involves the processing of an acquired image, including home or personal desktop scanning. Particular examples abound in the medical imaging field where several modalities, such as magnetic resonance, X-ray, ultrasound, and other techniques, are available for producing the data represented by the pixels. Depending upon the particular modality employed, the pixel data is detected and encoded, such as in the form of digital values. The values are linked to particular relative locations of the pixels in the reconstructed image.

20

25

The utility of a processed image is often largely dependent upon the degree to which it can be interpreted by users or processed by subsequent automation. For example, in the field of medical diagnostics imaging, MRI, X-ray, and other images are most useful when they can be easily understood and compared by an attending physician or radiologist. Other examples include biometric analysis where the processed image, such as a cornea or fingerprint, is often further processed by software or hardware based matching algorithms. Typically one impediment to

30

interpretation or further processing is the pixel to pixel variation which is attributable to acquisition noise. While acquisition noise is usually random, there may also be additional structured noise as well which may be observed as artifacts in the image. To mitigate the effects of random noise, many forms of noise reduction filters to improve the final image presentation have been proposed.

Moreover, while a number of image processing parameters may control the final image presentation, it is often difficult to determine which of these parameters, or which combination of the parameters, may be adjusted to provide the optimal image presentation. Often, the image processing techniques must be adjusted in accordance with empirical feedback from an operator, such as a physician or technician.

The facility with which a reconstructed discrete pixel image may be interpreted by an observer may rely upon intuitive factors of which the observer may not be consciously aware. For example, in medical imaging, a physician or radiologist may seek specific structures or specific features in an image such as bone, soft tissue or fluids. Such structures may be physically defined in the image by contiguous edges, contrast, texture, and so forth. Other forms of imaging, such as biometric analysis or optical character recognition, require the identification of specific structures or features, such as ridges or lines, which are similarly defined by contiguous edges and so forth.

The presentation of such features often depends heavily upon the particular image processing technique employed for converting the detected values representative of each pixel to modified values used in the final image. The image processing technique employed can therefore greatly affect the ability of an observer or an analytical device to recognize salient features of interest. The technique should carefully maintain recognizable structures of interest, as well as abnormal or unusual structures, while providing adequate textural and contrast information for interpretation of these structures and surrounding background. Ideally the technique

will perform these functions in a computationally efficient manner so that processing times, as well as hardware requirements, can be minimized.

Known signal processing systems for enhancing discrete pixel images suffer from certain drawbacks. For example, such systems may not consistently provide comparable image presentations in which salient features or structures may be easily visualized. Differences in the reconstructed images may result from particularities of individual scanners and circuitry, as well as from variations in the detected parameters (e.g. molecular excitation or received radiation). Differences can also result from the size, composition, position, or orientation of a subject or item being scanned. Signal processing techniques employed in known systems are often difficult to reconfigure or adjust, owing to the relative inflexibility of hardware or firmware devices in which they are implemented or to the coding approach employed in software.

Moreover, certain known techniques for image enhancement may offer excellent results for certain systems, but may not be as suitable for others. For example, in medical imaging, low, medium and high field MRI systems may require substantially different data processing due to the different nature of the data defining the resulting images. In current techniques completely different image enhancement frameworks are employed in such cases. In addition, current techniques may result in highlighting of small, isolated areas which are not important for interpretation and may be distracting to the viewer. Conversely, in techniques enhancing images by feature structure recognition, breaks or discontinuities may be created between separate structural portions, such as along edges. Such techniques may provide some degree of smoothing or edge enhancement, but may not provide satisfactory retention of textures at ends of edges or lines.

Finally, known signal processing techniques often employ computational noise reduction algorithms which are not particularly efficient, resulting in delays in formulation of the reconstituted image or under-utilization of signal processing capabilities. More computationally efficient algorithms would allow both quicker

image display, perhaps even approaching the level of real time display for some modalities. Further, more computationally efficient noise reduction algorithms might reduce or eliminate hardware based noise reduction, making them more suitable for diagnostic imaging systems due to both the increased speed and the less stringent equipment requirements.

There is a need, therefore, for a more computationally efficient technique for enhancing discrete pixel images which addresses these concerns. Ideally such a technique would be robust in its implementation, allowing it to be used with any number of pixel imaging modalities with few, if any, modifications.

SUMMARY OF THE INVENTION

The invention provides an improved technique for enhancing discrete pixel images which is computationally efficient and which maintains image quality. The technique provides a means of combining multi-resolution decomposition (wavelet based processing) with segmentation based techniques which identify structures within an image and separately process the pixels associated with the structures. This combination allows the technique to exploit the redundancies of an image, as with wavelet based techniques, while also allowing the separate processing of structures and non-structures, as in segmentation-based techniques. The combination of these techniques results in a computationally efficient, yet robust, noise reduction filter which may be applied to a variety of pixel based images.

Because of the efficiency of the technique, real-time or near real-time imaging may be performed in some diagnostic systems, such as ultrasound, without the hardware based noise reduction techniques which currently result in degraded, inferior images. Further, in other types of diagnostic imaging systems, the technique will allow for virtual dose improvements since a subject can be exposed to lower levels of a radiation source while still achieving the same quality of images. Any image acquisition system, however, may benefit from the application of the present technique.

In an exemplary embodiment, multi-resolution decomposition occurs when an image is shrunk by a given factor, allowing for the exploitation of redundancies in the image during subsequent processing. The shrunk image is then processed using segmentation based techniques which begin by identifying the structure elements within a blurred or smoothed image. Segmentation processing renders the structural details more robust and less susceptible to noise. A scalable threshold may serve as the basis for the identification of structural regions, making the enhancement framework inherently applicable to a range of image types and data characteristics. While small, isolated regions may be filtered out of the image, certain of these may be recuperated to maintain edge and feature continuity.

Following identification of the structures, portions of the image, including structural and non-structural regions, are smoothed. Structural regions may be smoothed to enhance structural features in dominant orientations, which may be identified to provide consistency both along and across structures. Non-structural regions may be homogenized to provide an understandable background for the salient structures. The structures may be further sharpened, and minute regions may be identified which are considered representative of noise. Such artifacts may be smoothed, removed or selectively sharpened based upon a predetermined threshold value.

Upon completion of the segmentation based processing, the image is expanded by the same factor it was originally shrunk by to return it to its original size. Original texture may be added back to non-structural regions to further facilitate interpretation of both the non-structural and structural features. The ability of the present technique to increase computational efficiency, due to exploitation of the image redundancies, while maintaining image quality is particularly noteworthy since a reduction in image quality might be expected as a result of the image resizing. Surprisingly, however, no such decrease in image quality is observed.

The technique is useful in processing any pixel based images. The technique is particularly useful in the context of a variety of medical imaging modalities, including magnetic resonance imaging, X-ray, CT, and ultrasound.

5

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 is a diagrammatical representation of an imaging system adapted to enhance discrete pixel images of a subject;

Fig. 2 is a diagram of an exemplary discrete pixel image made up of a matrix of pixels having varying intensities defining structures and non-structures;

10

Fig. 3 is a flow chart illustrating the progression of an image through multi-resolution decomposition and segmentation based processing.

Fig. 4 is a flow chart illustrating steps in exemplary control logic for multi-resolution decomposition of a discrete pixel image, for identification of structures, and for enhancement of both structural and non-structural regions in the image;

15

Fig. 5 is a flow chart illustrating steps in exemplary control logic for identifying structural features in a discrete pixel image;

Fig. 6 is a diagram of elements or modules used in the steps of Fig. 4 for generating gradient components for each discrete pixel of the image;

20

Fig. 7 is a gradient histogram of an image used to identify gradient thresholds for dividing structure from non-structure in the image;

Fig. 8 is a flow chart of steps in exemplary control logic for selectively eliminating small or noisy regions from the structure definition;

Fig. 9 is a flow chart of steps in exemplary control logic for processing structural features identified in the image by binary rank order filtering;

25

Fig. 10 is a flow chart illustrating steps in exemplary control logic for orientation smoothing of structure identified in an image;

Fig. 11 is a flow chart illustrating steps in exemplary control logic for performing dominant orientation smoothing in the process summarized in Fig. 9;

30

Fig. 12 is a diagram of directional indices employed in the orientation smoothing process of Fig. 10;

Fig. 13 is a flow chart of steps in exemplary control logic for performing local orientation smoothing through the process of Fig. 9;

Fig. 14 is a flow chart of steps in exemplary control logic for homogenization smoothing of non-structural regions of a discrete pixel image;

Fig. 15 is a flow chart of steps in exemplary control logic for orientation sharpening of structural regions in a discrete pixel image; and

Fig. 16 is a flow chart illustrating steps in exemplary control logic for reintroducing certain textural features of non-structural regions in a discrete pixel image.

DETAILED DESCRIPTION OF THE INVENTION

A highly abstracted rendition of image processing by the present technique is illustrated in Fig. 3, beginning with the input of the raw signal data as input image 70. Input image 70 is shrunk by a user configurable parameter, X , to create shrunk image 72. Shrunk image 72 undergoes normalization to create normalized image 74. Threshold criteria are applied to identify structures within normalized image 74. The structures identified are used to generate a structure mask 76 which is used in subsequent processing to distinguish both structure and non-structure regions, allowing differential processing of these regions. Normalized image 74 is filtered to reduce noise via structure mask 76 to create an intermediate filtered image 78 which is subsequently normalized to form renormalized image 80. Renormalized image 80 and structure mask 76 are expanded to form expanded image 82 and expanded structure mask 83. Differential blending of expanded image 82 and input image 70 is accomplished via the application of expanded structure mask 83. The product of the blending process is final image 84. More particular descriptions of this technique follows.

Referring to Fig. 1, an image acquisition system 10 is illustrated as including a scanner 12 coupled to circuitry for acquiring and processing discrete pixel data. Scanner 12 may be a variety of different scanning modalities including medical imaging modalities such as ultrasound, CT, MR, X-ray, fluoroscopy, CR, and PET.

Signals sensed by scanner 12 are encoded to provide digital values representative of the signals associated with specific locations on or in the subject, and are transmitted to image acquisition circuitry 22. Image acquisition circuitry 22 also provides control signals for configuration and coordination of scanner operation during image acquisition. Image acquisition circuitry 22 transmits the encoded image signals to an image processing circuit 24. Image processing circuit 24 executes pre-established control logic routines stored within a memory circuit 26 to filter and condition the signals received from image acquisition circuitry 22 to provide digital values representative of each pixel in the acquired image. These values are then stored in memory circuit 26 for subsequent processing and display. Alternately, image acquisition circuitry 22 may transmit the encoded image signals to memory circuit 26. Image processing circuit 24 may subsequently acquire the signals from memory circuit 26 for the filtering and conditioning steps described above.

Image processing circuit 24 receives configuration and control commands from an input device 28 via an input interface circuit 30. Input device 28 will typically include an operator's station and keyboard for selectively inputting configuration parameters and for commanding specific image acquisition sequences. Image processing circuit 24 is also coupled to an output device 32 via an output interface circuit 34. Output device 32 will typically include a monitor or printer for generating reconstituted images based upon the image enhancement processing carried out by circuit 24.

It should be noted that the signal processing techniques described herein are not limited to any particular imaging modality. Accordingly, these techniques may be applied to image data acquired by magnetic resonance imaging, X-ray systems, PET systems, and computer tomography systems, among others. It should also be noted that in the embodiment described, image processing circuit 24, memory circuit 26, and input and output interface circuits 30 and 34 are included in a programmed digital computer. However, circuitry for carrying out the techniques described herein may be configured as appropriate coding in application-specific microprocessors, analog circuitry, or a combination of digital and analog circuitry.

Fig. 2 illustrates an exemplary discrete pixel image 50 produced via system 10. Image 50 is composed of a matrix of discrete pixels 52 disposed adjacent to one another in a series of rows 54 and columns 56. These rows and columns of pixels provide a pre-established matrix width 58 and matrix height 60. Typical matrix dimensions may include 256 X 256 pixels; 512 X 512 pixels; 1,024 X 1,024 pixels, and so forth. The particular image matrix size may be selected via input device 28 (see Fig. 1) and may vary depending upon such factors as the subject to be imaged and the resolution desired.

Illustrated in Fig. 2, exemplary image 50 includes structural regions 62, illustrated as consisting of long, contiguous lines defined by adjacent pixels. Image 50 also includes non-structural regions 64 lying outside of structural regions 62. Image 50 may also include isolated artifacts 66 of various sizes (i.e., number of adjacent pixels), which may be defined as structural regions, or which may be eliminated from the definition of structure in accordance with the techniques described below. It should be understood that the structures and features of exemplary image 50 are also features of the specific and modified images discussed above in relation to Fig. 3.

Structural regions 62 and non-structural regions 64 are identified and enhanced in accordance with control logic summarized generally in Fig. 4. This control logic is preferably implemented by image processing circuit 24 based upon appropriate programming code stored within memory circuit 26. The control logic routine, designated generally by reference numeral 120 in Fig. 4, begins at step 121 with the initialization of parameters employed in the image enhancement process. This initialization step includes the reading of default and operator-selected values for parameters described in the following discussion, such as the size of small regions to be eliminated from structure, a "focus parameter" and so forth. Where desired, certain of these parameters may be prompted via input device 28, requiring the operator to select between several parameter choices, such as image matrix size.

Next, at step 123, image processing circuit 24 collects the raw image data 70, represented as I_{raw} and shrinks the image. The shrinking of step 123 may be accomplished by various sub-sampling techniques, including pixel averaging, which read the digital values representative of intensities at each pixel and then shrink the image by some factor X which is generally greater than one. In the preferred embodiment, a 2x2 or 3x3 boxcar filter may be applied to obtain a non-overlapping average. Multi-dimensional factors may also be employed, such as 2x3 or 3x2 filters. A multi-dimensional factor must be greater than one in at least one of the dimensions, such as 3x1 or 1x3. In order to obtain a non-overlapping average, pixels may be mirrored at the boundaries when needed. A shrunk image 72, I_{shrunk} , is the product of the sub-sampling technique.

At step 124, image processing circuit 24 normalizes the image values acquired for the pixels defining the shrunk image 72. In the illustrated embodiment, this step includes reading digital values representative of intensities at each pixel, and scaling these intensities values over a desired dynamic range. For example, the maximum and minimum intensity values in the image may be determined, and used to develop a scaling factor over the full dynamic range of output device 32. Moreover, a data offset value may be added to or subtracted from each pixel value to correct for intensity shifts in the acquired data. Thus, at step 124, circuit 24 processes I_{shrunk} in Fig. 4, to produce normalized image 74, I_{normal} . Normalized image 74 includes pixel values filtered to span a desired portion of a dynamic range, such as 12 bits, independent of variations in the acquisition circuitry or subject.

It should be noted that while reference is made in the present discussion to intensity values within an image, such as input image 70, shrunk image 72, normalized image 73, or exemplary image 50, the present technique may also be used to process other parameters encoded for individual pixels 52 of an image. Such parameters might include frequency or color, not merely intensity.

At step 126, image processing circuit 24 executes a predetermined logic routine for identifying structure 62 within normalized image 74, as defined by data representative of the individual pixels of the image. Exemplary steps for identifying the structure in accordance with the present technique are described below with reference to Fig. 5. The structure identified at step 126 is used to generate a structure mask 76, $M_{\text{structure}}$, which is used in subsequent steps. Step 128 uses structure mask 76 to identify structure which is then orientation smoothed as summarized below with reference to Figs. 10-13. While various techniques may be employed for this orientation smoothing, in the embodiment described, dominant orientation smoothing may be carried out, which tends to bridge gaps between spans of structure, or local orientation smoothing may be employed to avoid such bridging. Orientation smoothing carried out in step 128 thus transforms normalized image 74 to a filtered image 78, I_{filtered} , which will be further refined by subsequent processing. After the structure identified at step 126 has been orientation smoothed, the structure regions are then orientation sharpened at step 132 to further refine filtered image 78. The process of orientation sharpening is described more fully below with reference to Fig. 15.

In parallel with the processing of the structure regions described in steps 128 and 132, the non-structure regions of normalized image 74 are further processed as follows to also contribute to filtered image 78. At step 130, image processing circuit 24 performs homogenization smoothing on non-structure regions of normalized image 74. As described more fully below with reference to Fig. 14, this homogenization smoothing is intended to blend features of non-structural regions into the environment surrounding the structure identified at step 126. At step 134 the filtered image 78 is then renormalized based upon the intensity values after filtering and the original normalized intensity range to produce renormalized image 80.

At step 135, both structure mask 76 and renormalized image 80 are expanded by the same factor by which raw image 70 was originally shrunk in step 123. The products of step 135 are thus an expanded structure mask 83 and expanded image 82, both of which are the same dimensions as input image 70. Finally, at step

136 texture present in input image 70 is blended back into the expanded image 82, I_{expanded} to provide texture for final image 84. The blending process of step 136 utilizes expanded structure mask 83 to allow differential blending of structure and non-structure regions. The texture blending process is described below with reference to Fig. 16. Following step 136, the resulting pixel image values are stored in memory circuit 26 for eventual reconstruction, display, or analysis as final image 84.

Fig. 5 illustrates steps in control logic for identifying structural regions 62 within normalized image 74 and for eliminating small or noisy isolated regions from the definition of the structural regions. As indicated above, the logic of Fig. 5, summarized as step 126 in Fig. 4, begins with pixel data of the normalized image 74.

At step 150 a blurred or smoothed version of normalized image 74 is preferably formed. It has been found that by beginning the steps of Fig. 5 with this smoothed image, structural components of the image may be rendered more robust and less susceptible to noise. While any suitable smoothing technique may be employed at step 150, in the present embodiment, a box-car smoothing technique is used, wherein a box-car filter smoothes the image by averaging the value of each pixel with values of neighboring pixels. As will be appreciated by those skilled in the art, a computationally efficient method for such filtering may be implemented, such as employing a separable kernel (3 or 5 pixels in length) which is moved horizontally and vertically along the image until each pixel has been processed.

At step 152, X and Y gradient components for each pixel are computed based upon the smoothed version of normalized image 74. While several techniques may be employed for this purpose, in the presently preferred embodiment, 3 X 3 Sobel modules or operators 180 and 182, illustrated in Fig. 6, are employed. As will be appreciated by those skilled in the art, module 180 is used for identifying the X gradient component, while module 182 is used for identifying the Y gradient component of each pixel. In this process, modules 180 and 182 are superimposed

over the individual pixel of interest, with the pixel of interest situated at the central position of the 3 X 3 module. The intensity values located at the element locations within each module are multiplied by the scalar value contained in the corresponding element, and the resulting values are summed to arrive at the corresponding X and Y gradient components.

With these gradient components thus computed, at step 154 the gradient magnitude, Gmag, and gradient direction, Gdir, are computed. In the presently preferred technique, the gradient magnitude for each pixel is equal to the higher of the absolute values of the X and Y gradient components for the respective pixel. The gradient direction is determined by finding the Arctangent of the Y component divided by the X component. For pixels having an X component equal to zero, the gradient direction is assigned a value of $\pi/2$. The values of the gradient magnitudes and gradient directions for each pixel are saved in memory circuit 26.

It should be noted that alternative techniques may be employed for identifying the X and Y gradient components and for computing the gradient magnitudes and directions. For example, those skilled in the art will recognize that in place of the Sobel gradient modules 180 and 182, other modules such as the Roberts or Prewitt operators may be employed. Moreover, the gradient magnitude may be assigned in other manners, such as a value equal to the sum of the absolute values of the X and Y gradient components.

Based upon the gradient magnitude values determined at step 154, a gradient histogram is generated as indicated at step 156. Fig. 7 illustrates an exemplary gradient histogram of this type. The histogram, designated by reference numeral 190, is a bar plot of specific populations of pixels having specific gradient values. These gradient values are indicated by positions along a horizontal axis 192, while counts of the pixel populations for each value are indicated along a vertical axis 194, with each count falling at a discrete level 196. The resulting bar graph forms a step-wise gradient distribution curve 198. Those skilled in the art will appreciate that in the actual implementation the histogram of Fig. 7 need not be represented

graphically, but may be functionally determined by the image processing circuitry operating in cooperation with values stored in memory circuitry.

Histogram 190 is used to identify a gradient threshold value for separating structural components of the image from non-structural components. The threshold value is set at a desired gradient magnitude level. Pixels having gradient magnitudes at or above the threshold value are considered to meet a first criterion for defining structure in the image, while pixels having gradient magnitudes lower than the threshold value are initially considered non-structure. The threshold value used to separate structure from non-structure is preferably set by an automatic processing or "autofocus" routine as defined below. However, it should be noted that the threshold value may also be set by operator intervention (e.g. via input device 28) or the automatic value identified through the process described below may be overridden by the operator to provide specific information in the resulting image.

As summarized in Fig. 5, the process for identification of the threshold value begins at step 158 by selecting an initial gradient threshold. This initial gradient threshold, designated 200 in Fig. 7 is conveniently set to a value corresponding to a percentile of the global pixel population, such as 30 percent. The location along axis 192 of the IGT value 200 is thus determined by adding pixel population counts from the left-hand edge of histogram 190 of Fig. 7, adjacent to axis 194 and moving toward the right (i.e., ascending in gradient values). Once the desired percentile value is reached, the corresponding gradient magnitude is the value assigned to the IGT.

At step 160, a search is performed for edges of the desired structure. The edge search proceeds by locating the pixels having gradient magnitudes greater than the IGT value selected in step 158 and considering a 5 X 5 pixel neighborhood surrounding the relevant pixels of interest. Within the 5 X 5 pixel neighborhood of each pixel of interest, pixels having gradient magnitudes above the IGT and having directions which do not differ from the direction of the pixel of interest by more than a predetermined angle are counted. In the presently preferred embodiment, an angle

of 0.35 radians is used in this comparison step. If the 5 X 5 neighborhood count is greater than a preset number, 3 in the present embodiment, the pixel of interest is identified as a relevant edge pixel. At step 162, a binary mask image is created wherein pixels identified as relevant edge pixels in step 160 are assigned a value of 1, while all other pixels are assigned a value equal to zero.

At step 164 small or noisy segments identified as potential candidates for structure are iteratively eliminated. Steps in control logic for eliminating these segments are summarized in Fig. 8. Referring to Fig. 8, the process begins at step 210 where a binary image is obtained by assigning a value of 1 to pixels having a gradient magnitude value equal to or greater than a desired value, and a value of zero to all other pixels. This binary image or mask is substantially identical to that produced at step 162 (see Fig. 5). At step 212 each pixel having a value of 1 in the binary mask is assigned an index number beginning with the upper-left hand corner of the image and proceeding to the lower right. The index numbers are incremented for each pixel having a value of 1 in the mask. At step 214 the mask is analyzed row-by-row beginning in the upper left by comparing the index values of pixels within small neighborhoods. For example, when a pixel is identified having an index number, a four-connected comparison is carried out, wherein the index number of the pixel of interest is compared to index numbers, if any, for pixels immediately above, below, to the left, and to the right of the pixel of interest. The index numbers for each of the connected pixels are then changed to the lowest index number in the connected neighborhood. The search, comparison and reassignment then continues through the entire pixel matrix, resulting in regions of neighboring pixels being assigned common index numbers. In the preferred embodiment the index number merging step of 214 may be executed several times, as indicated by step 216 in Fig. 8. Each subsequent iteration is preferably performed in an opposite direction (i.e., from top-to-bottom, and from bottom-to-top).

Following the iterations accomplished through subsequent search and merger of index numbers, the index number pixel matrix will contain contiguous regions of pixels having common index numbers. As indicated at step 218 in Fig. 8,

a histogram is then generated from this index matrix by counting the number of pixels having each index number appearing in the index matrix. As will be apparent to those skilled in the art, each separate contiguous region of pixels having index numbers will have a unique index number. At step 220, regions represented by index numbers having populations lower than a desired threshold are eliminated from the definition of structure as determined at step 162 of Fig. 5. In a presently preferred embodiment, regions having a pixel count lower than 50 pixels are eliminated in step 220. The number of pixels to be eliminated in this step, however, may be selected as a function of the matrix size, and the amount and size of isolated artifacts to be permitted in the definition of structure in the final image.

Returning to Fig. 5, with pixels for small segments eliminated from the binary mask created at step 162, the number of pixels remaining in the binary mask are counted as indicated at step 166. While the resulting number may be used to determine a final gradient threshold, it has been found that a convenient method for determining a final gradient threshold for the definition of structure includes the addition of a desired number of pixels to the resulting pixel count. For example, in a presently preferred embodiment a value of 4,000 is added to the binary mask count resulting from step 164 to arrive at a desired number of pixels in the image structure definition. This parameter may be set as a default value, or may be modified by an operator. In general, a higher additive value produces a sharper image, while a lower additive value produces a smoother image. This parameter, referred to in the present embodiment as the “focus parameter” may thus be varied to redefine the classification of pixels into structures and non-structures.

With the desired number of structure pixels thus identified, a final gradient threshold or FGT is determined as illustrated at step 168 in Fig. 5, based upon the histogram 190 as shown in Fig. 7. In particular, the population counts for each gradient magnitude value beginning from the right-hand edge of histogram 190 are summed moving to the left as indicated by reference number 202. Once the desired number of structural pixels is reached (i.e., the number of pixels counted at step 166 plus the focus parameter), the corresponding gradient magnitude value is identified

as the final gradient threshold 204. In the presently preferred embodiment, the FGT value is then scaled by multiplication by a value which may be automatically determined or which may be set by a user. For example, a value of 1.9 may be employed for scaling the FGT, depending upon the image characteristics, the type and features of the structure viewable in the image, and so forth. The use of a scalable threshold value also enables the technique to be adapted easily and quickly to various types of images, such as for MRI data generated in systems with different field strengths, CT data, and so forth.

Based upon this scaled final gradient threshold, a new binary mask is defined by assigning pixels having values equal to or greater than the FGT a value of 1, and all other pixels a value of zero. At step 170 the resulting binary mask is filtered to eliminate small, isolated segments in a process identical to that described above with respect to step 164 and Fig. 8. However, at step 170 rather than a four-connected neighborhood, a eight-connected neighborhood (i.e., including pixels having shared edges and corners bounding the pixel of interest) is considered in the index number merger steps.

At step 172, certain of the isolated regions may be recuperated to provide continuity of edges and structures. In the present embodiment, for example, if a pixel in the gradient image is above a second gradient threshold, referred to as GFT, and is connected (i.e. immediately adjacent) to a pixel which is above the FGT, the corresponding pixel in the binary image is changed from a 0 value to a value of 1. The value of the GFT may be set to a desired percentage of the FGT, and may be determined empirically to provide the desired degree of edge and structure continuity. This gradient following step is preferably carried out recursively to determine an initial classification of the pixels.

At step 174 in Fig. 5, the feature edges identified through the previous steps, representative of candidate structures in the image, are binary rank order filtered. While various techniques may be employed for this enhancing identified candidate structures, it has been found that the binary rank order filtering provides satisfactory

results in expanding and defining the appropriate width of contiguous features used to define structural elements. Steps in exemplary control logic for implementing the binary rank order filtering of step 174 are illustrated in Fig. 9.

5 Referring to Fig. 9, the binary rank order filtering begins at step 230 with the binary mask generated and refined in the foregoing steps. At step 230, circuit 24 determines whether each pixel in the binary mask has a value of 1. If the pixel found to have a value of 1 in the mask, a neighborhood count is performed at step 232. In this neighborhood count, pixels in the binary mask having values of 1 are
 10 counted within a 3 X 3 neighborhood surrounding the structural pixel of interest. This count includes the pixel of interest. At step 234, circuit 24 determines whether the count from step 232 exceeds a desired count m . In the present embodiment, the value of m used at step 234 is 2. If the count is found to exceed the value m the value of 1 is reassigned to the pixel of interest, as indicated at step 236. If, however,
 15 the count is found not to exceed the value of m the pixel of interest is assigned the value of 0 in the mask as indicated at step 238. Following steps 236 and 238, or if the pixel is found not to have an original value of 1 in the mask at step 230, control proceeds to step 240.

20 At step 240, circuit 24 reviews the structure mask to determine whether each pixel of interest has a value of 0. If a pixel is located having a value of 0, circuit 24 advances to step 242 to compute a neighborhood count similar to that described above with respect to step 232. In particular, a 3 X 3 neighborhood around the non-structure pixel of interest is examined and a count is determined of pixels in that
 25 neighborhood having a mask value of 1. At step 244 this neighborhood count is compared to a parameter n . If the count is found to exceed the parameter n , the mask value for the pixel is changed to 1 at step 246. If the value is found not to exceed n , the mask pixel retains its 0 value as indicated at step 248. In the present embodiment, the value of n used in step 244 is 2. Following step 246 or step 248,
 30 the resulting structure mask 76, $M_{\text{structure}}$, contains information identifying structural features of interest and non-structural regions. Specifically, pixels in structure mask

76 having a value of 1 are considered to identify structure, while pixels having a value of 0 are considered to indicate non-structure.

With the structure of the image thus identified, orientation smoothing of the structure, as indicated at step 128 of Fig. 4, is carried out through logic such as that illustrated diagrammatically in Fig. 10. As shown in Fig. 10, the orientation smoothing of image structure begins by looking at the structure pixels of normalized image 74, as determined via application of structure mask 76, and may proceed in different manners depending upon the type of smoothing desired. In particular, based upon an operator input designated 260 in Fig. 10, a logical decision block 262 directs image processing circuit 24 to either dominant orientation smoothing as indicated at reference numeral 264 or local orientation smoothing as indicated at 266. If dominant orientation smoothing is selected, the intensity values for the structural pixels are processed as summarized below with respect to Fig. 11, to generate a binary mask M'. Following iterations of the procedure outlined below with reference to Fig. 11, the values of mask M' are evaluated at step 268, and smoothing is performed on the structure intensity values by use of multipliers α and β resulting in values which are then summed as indicated at blocks 270, 272 and 274 of Fig. 10 and as summarized in greater detail below.

To explain the dominant orientation smoothing step of 264, reference is now made to Fig. 11. As illustrated in Fig. 11, the dominant orientation smoothing begins with assigning directional indices to each pixel identified as a structural pixel in structure mask 76. In the present embodiment, one of four directional indices is assigned to each structural pixel in accordance with the statistical variances for each pixel, as shown in Fig. 12. As illustrated in Fig. 12, within a local neighborhood 300 surrounding each structural pixel, statistical variances for pixel kernels in four directions are computed by reference to the normalized intensity values of the surrounding pixels. The direction of the minimum variance is selected from the four computed values and a corresponding directional index is assigned as indicated by reference numeral 302 in Fig. 12. In the present embodiment these directional indices are assigned as follows: "1" for 45 degrees; "2" for 135 degrees; "3" for 90

degrees; and “4” for 0 degrees. These steps are summarized as 282 and 284 in Fig. 11. At step 286 a local area threshold value is assigned based upon the image matrix size. In the present embodiment, a local area threshold of 6 is used for 256 X 256 pixel images, a value of 14.25 is used for 512 X 512 pixel images, and a value of 23 is used for 1024 X 1024 pixel images.

At step 288, a binary mask M' is initialized with zero values for each pixel. At step 290 a dominant orientation is established for each structural pixel by examining the directional indices set in step 284 within a local neighborhood surrounding each structural pixel. In this process, the directional indices found in the local neighborhood are counted and the pixel of interest is assigned the directional index obtaining the greatest count (or the lowest index located in the case of equal counts).

In the present embodiment, both the dominant direction and its orthogonal direction are considered to make a consistency decision in the dominant orientation smoothing operation. In terms of Fig. 12, these directions are 1 and 2, or 3 and 4. It has been found that considering such factors substantially improves the robustness of the dominant orientation determination in the sense of being consistent with the human visual system (i.e. providing reconstructed images which are intuitively satisfactory for the viewer).

The consistency decision made at step 290 may be based upon a number of criteria. In the present embodiment, the image is smoothed along the dominant direction (i.e. the direction obtaining the greatest number of counts in the neighborhood) if any one of the following criteria is met: (1) the number of counts of the orientation obtaining the greatest number is greater than a percentage (e.g. 67%) of the total neighborhood counts, and the orthogonal orientation obtains the least counts; (2) the number of counts of the orientation obtaining the maximum counts is greater than a smaller percentage than in criterion (1) (e.g. 44%) of the total neighborhood counts, and the orthogonal direction obtains the minimum number, and the ratio of the counts of the dominant direction and its orthogonal is greater

than a specified scalar (e.g. 5); or (3) the ratio of the dominant direction counts to its orthogonal direction counts is greater than a desired scalar multiple (e.g. 10).

In the present embodiment, the neighborhood size used to identify the direction of dominant orientation in step 290 is different for the series of image matrix dimensions considered. In particular, a 3 X 3 neighborhood is used for 256 X 256 pixel images, a 5 X 5 neighborhood is used for 512 X 512 pixel images, and a 9 X 9 neighborhood is used for 1024 X 1024 pixel images.

At step 292, the count determined in the searched neighborhood for each pixel is compared to the local area threshold. If the count is found to exceed the local area threshold, image processing circuit 24 advances to step 294. At that step, the intensity value for each structural pixel is set equal to the average intensity of a 1 X 3 kernel of pixels in the dominant direction for the pixel of interest. Subsequently, at step 296, the value of a corresponding location in the binary matrix M' is changed from 0 to 1. If at step 292, the count is found not to exceed the local area threshold for a particular pixel, the intensity value for the pixel of interest is set equal to a weighted average as indicated at step 298. This weighted average is determined by the relationship:

$$\text{weighted avg} = (1/1+p)(\text{input}) + (p/1+p)(\text{smoothed value});$$

where the *input* value is the value for the pixel of interest at the beginning of routine 264, *p* is a weighting factor between 1 and 200, and the *smoothed value* is the average intensity of a 1 X 3 kernel in the dominant direction of the pixel of interest. From either step 296 or 298, circuit 24 returns to step 268 of Fig. 10.

Referring again to Fig. 10, at step 268, the values of each pixel in the binary mask M' are evaluated. If the value is found to equal zero, the corresponding intensity value is multiplied by a weighting factor α at step 270. In the present embodiment, factor α is set equal to 0.45. At block 272 the resulting value is summed with the product of the normalized intensity value for the corresponding

pixel and a weighting factor β as computed at step 274. In the present embodiment, the factors α and β have a sum equal to unity, resulting in a value of β equal to 0.55.

If at step 268 the value for a particular pixel is found to equal 1 in the binary mask M' , control advances to decision block 276. Decision block 276 is also reached following the summation performed at block 272 as described above. In the present embodiment, the foregoing dominant orientation smoothing steps are performed over a desired number of iterations to provide sufficient smoothing and bridging between structural regions. At step 276, therefore, circuit 24 determines whether the desired number of iterations have been completed, and if not, returns to step 264 to further smooth the structural regions. In the present embodiment, the operator may select from 1 to 10 such iterations.

As noted above, the orientation smoothing can proceed through an alternative sequence of steps for local orientation smoothing as noted at block 266 in Fig. 10. Fig. 13 illustrates exemplary steps in control logic for such local orientation smoothing. As with the dominant orientation smoothing, the local orientation smoothing begins with the normalized intensity values for the structural pixels. At step 310, statistical variances for 1 X 3 pixel kernels about each structural pixel are calculated for each indexed direction (see Fig. 12) as described above for the dominant orientation smoothing process. At step 312, a ratio of the maximum/minimum statistical variances identified for each pixel in step 310 is computed. At step 314 this ratio for each structural pixel is compared to a parameter R , referred to as a relaxation factor for the local orientation filtering. In the present embodiment, the value of R can be set between 1 and 200. If at step 314 the variance ratio is found to exceed R , local orientation filtering is accomplished as indicated at step 316 by setting the intensity value for the structural pixel of interest equal to an average value for the 1 X 3 pixel kernel in the direction of the minimum variance. If at step 314 the ratio between the maximum and minimum variances for the pixel of interest is found not to exceed R , no local orientation smoothing is performed and circuit 24 advances to a point beyond step 316. From this point, control returns to block 270 of Fig. 10.

As illustrated in Fig. 10, at block 270 the intensity value for each structural pixel is multiplied by a weighting factor α , and combined at block 272 with the product of the normalized intensity value for the corresponding pixel and a weighting factor β produced at block 274. As summarized above, at step 276,
 5 circuit 24 determines whether the desired number of iterations has been completed and, if not, returns to the local orientation smoothing block 266, to repeat the steps of Fig. 13 until the desired number of iterations is complete. Once the desired iterations have been performed, the filtered image 78 resulting from the orientation smoothing is further filtered by the processes described below.

10 As summarized above with reference to Fig. 4, in parallel with orientation smoothing of the structure identified within the image, homogenization smoothing of non-structure is performed. The steps in a process for such homogenization smoothing are summarized in Fig. 14. As shown in Fig. 14, the normalized intensity
 15 values for non-structural pixels are considered in this process. At step 330, the mean neighborhood intensity value for each non-structural pixel is computed (taking into account the normalized values of structural pixels where these are included in the neighborhood considered). In the present embodiment, step 330 proceeds on the basis of a 3 X 3 neighborhood surrounding each non-structural pixel. This mean
 20 value is assigned to the pixel of interest and control advances to step 334. At step 334 circuit 24 determines whether a desired number of iterations has been completed. If not, control returns to step 330 for further homogenization of the non-structural pixel intensity values. Once the desired number of iterations has been completed the homogenization smoothing routine of Fig. 14 is exited. In the present
 25 embodiment, the operator may set the number of homogenization smoothing iterations from a range of 1 to 10.

The filtered image is further processed by orientation sharpening of the identified structure pixels as mentioned above with regard to Fig. 4, and as
 30 illustrated in greater detail in Fig. 15. As shown in Fig. 15, in the present embodiment, the sharpening is performed only for pixel values which are above a preset lower limit, as indicted at decision block 340. This limit, which may be set to

a multiple of the FGT (e.g. 2 X FGT), thus avoids enhancement of structural pixels which should not be sharpened. If a structural pixel has a value above the limit, the orientation sharpening sequence begins at step 342 where Laplacian values for each such structural pixel are computed in the indexed directions shown in Fig. 12 and described above. The Laplacian values may be computed from the formula

$$E(k) = 2.0 * I(k) - I(k-1) - I(k+1);$$

where k is the structural pixel of interest, “k-1” is the pixel preceding the pixel of interest in the indexed direction, and “k+1” is the pixel succeeding the pixel of interest in the indexed direction. E(k) is the edge strength and I(k) is the intensity value at the structural pixel of interest. It should be noted that the Laplacian values computed at step 342 are based upon the filtered intensity values (i.e., smoothed values for structure). At step 344, the maximum of the four Laplacian values for each structural pixel is then saved to form an edge mask, M_{edge} . In forming M_{edge} border pixels in a given image are set to 0 for the subsequent steps.

At step 346, for each structural pixel of M_{edge} , the statistical variances and mean values for a 3 X 1 pixel kernel are computed in the indexed directions shown in Fig. 12, again using the filtered (i.e., homogenized and smoothed) values for each pixel in the relevant neighborhoods. The direction of minimum variance for each structural pixel is then identified from these values, and the mean value in the direction of minimum variance is saved for each pixel as indicated at step 348 to form a smoothed edge mask, $M_{smooth\ edge}$. At step 350, the mean value in the direction of minimum variance for each structural pixel of $M_{smooth\ edge}$ is multiplied by a configurable parameter. In the present embodiment, the value of the configurable parameter may be set to any number greater than 0 depending on the application. In general, the higher the value of the configurable parameter selected, the greater the overall sharpness of strong edges in the final image.

At step 351, each pixel, after multiplication, is compared to both a minimum and a maximum threshold value. Pixels which exceed the maximum threshold value

are set equal to the maximum threshold value. Likewise, pixels which are less than the minimum threshold value are set equal to the minimum threshold value. At step 353, the resulting weighted values, represented in $M_{\text{smooth edge}}$, are added to the initial filtered values for the corresponding structural pixel to form a new filtered image 78.

5 If the resulting intensity for the structural pixel is less than 0, its intensity is set to 0. In the present preferred embodiment, if the resulting intensity for the structural pixel exceeds 4,095, its intensity is set to 4,095. This upper limit is configurable to any number greater than 0. The effect of the aforementioned operations is to more strongly enhance weaker edges while providing a more limited enhancement to

10 edges which are already strong. The resulting filtered image values are then further processed as described below.

Following orientation sharpening of the structural features of the image and homogenization smoothing of non-structure regions, the entire image is again

15 renormalized as indicated at step 134 in Fig. 4. While various methods may be used for this renormalization, in the present embodiment the global average pixel intensity in filtered image 78 following steps 130 and 132 is computed, and a normalization factor is determined based upon the difference between this average value and the average value prior to the filtration steps described above. The new

20 normalized intensity value for each pixel is then determined by multiplying this normalization factor by the filtered pixel intensity, and adding the global minimum intensity value from the original data to the product.

The resulting renormalized image 80, denoted I_{renormal} in Fig. 4, is then

25 expanded by the same factor, X , by which the input image 70 was shrunk. Structure mask 76 is also expanded by this time by the same factor. Various suitable interpolation techniques may be used to accomplish this expansion including cubic interpolation. The products of expansion step 135 are thus an expanded structure mask 83 and an expanded image 82, each with the same dimensions as original input

30 image 70.

Expanded image 82 is processed to blend texture from input image 70 into expanded image 82 via expanded structure mask 83, as can be seen in Fig. 4 at step 136. This texture blending step is summarized in Fig. 16. Expanded structure mask 83 allows texture blended with pixels from expanded image 82 to be weighted differently depending upon whether a pixel is defined as structure or non-structure.

In general, the steps of Fig. 16 tend to add more or less original texture depending upon the gradient magnitude of the pixels. In particular, at step 360, the gradient magnitude for each pixel of interest is compared to a threshold value T . In the present embodiment, this threshold is set to a value of 300. If the gradient is found not to exceed the threshold, the pixel intensity value is multiplied by a value “a” at step 362. The resulting product is added at step 364 to the product of the raw intensity value for the pixel (prior to the shrinking at step 123 of Fig. 4) multiplied by a value equal to “1-a” at step 366. The resulting weighted average is assigned to the pixel.

If at step 360, the gradient magnitude value for a pixel is found to exceed the threshold value T , the pixel intensity is multiplied by a factor “b”, as noted at step 368. The resulting product is then added at step 370 to the product of the raw intensity for that pixel and a multiplier equal to “1-b” determined at step 372. In the present embodiment, the value of “b” may be set within a range from 0 to 1, with the value of “a” being set equal to 1.5 times the value of “b”. As will be apparent to those skilled in the art, the weighted averaging performed by the steps summarized in Fig. 16 effectively adds texture to provide an understandable environment for the structures identified as described above. By performing the comparison at step 360, the process effectively adds less original texture for pixels having low gradient values, and more original texture for pixels having higher gradient values. Where desired, the values of “a” and “b” may be set so as to increase or decrease this function of the process. The product of the differential textural blending of step 136 is final filtered image 84 which may be displayed, stored in memory, or further processes and analyzed.